CIRP ICME 2010 - 7th CIRP International Conference on Intelligent Computation in Manufacturing Engineering : Innovative and Cognitive Production Technology and Systems. Naples, Italy. 23-25 June 2010. Ed. R. Teti

Business Process Mining for Industry. Successes and Caveats

J. Mehnen[1], C. Turner[1], A. Tiwari[1]
[1]Decision Engineering Centre, Cranfield University, UK

Abstract
Business Process Mining (BPM) is a powerful technique which aims at mapping the complex structure of industrial processes into human interpretable graph structures by analysing business process traces automatically. The transfer of an innovative idea into an industrially viable product is a challenging task in its own rights.
First, this paper introduces the concept of business process mining and an innovative Genetic Programming (GP) approach. Second, this paper addresses the principal caveats and solutions that come with transferring new academic solutions into real-world applications. A real BPM transfer project serves a background for this discussion.

Keywords:
Business Process Mining, Genetic Programming, Academic Knowledge Transfer, Industrial Case Study

1 INTRODUCTION

The modern globalised world gets more and more connected while more and advanced communication technology gets available. Business processes become more flexible but also more complex. Online banking via computer is a standard and banking via mobile phone becomes increasingly popular. Internet services are readily available and can be freely combined to generate new services. As processes become bigger and more dynamic they also become harder to manage.

This can create loopholes which can be exploited by fraudsters. Deviations from standard procedures can cause severe issues in hospitals, call centres or high tech production. Deviations from the standard can also be beneficial as they can be indicators of potentially inefficient processes. People may cleverly shortcut unnecessary and cumbersome procedures.

Analysis and visualisation of complex but highly automated processes as they appear in data rich environments such as in banks, insurances or high-tech manufactures is the domain of Business Process Mining (BPM). Process Mining (PM), as the overruling term, is a particular discipline in Data Mining. Data Mining is the application of a set of statistical tools for foraging for useful and actionable information in large data bases. Process Mining uses data mining techniques to analyse data traces of automated processes while BPM focuses on the narrower subset of business related process data.

Business Process Mining can be a valuable investment as business data and computer power becomes easily available. While Data Mining generally identifies typical properties of customers by e.g. inspecting the basket of commodities, the general Data Mining approach does not tell the whole story. It does not inform about how the customer reached his/her decision and what steps the customer followed. Data Mining is well established in fraud detection. However, also here general Data Mining often just identifies typical classes of fraudsters rather than how the actual fraud has been committed.
The identification of the actual process steps is the key business in Business Process Mining. BPM helps automatically layouting process maps that can be used to visualise and understand complex processes better. Therefore, BPM can be very useful in process analysis, process visualisation and process optimisation.

This paper introduces an innovative Business Process Mining technique, which is based on Genetic Programming. This idea is now in a state where it is ready for the market. However, the way from an idea and a software prototype to a marketable product is long. This paper discusses also the difficult way of new BPM software maturing from plain academic proof-of-concept-research to an industrially applicable product.

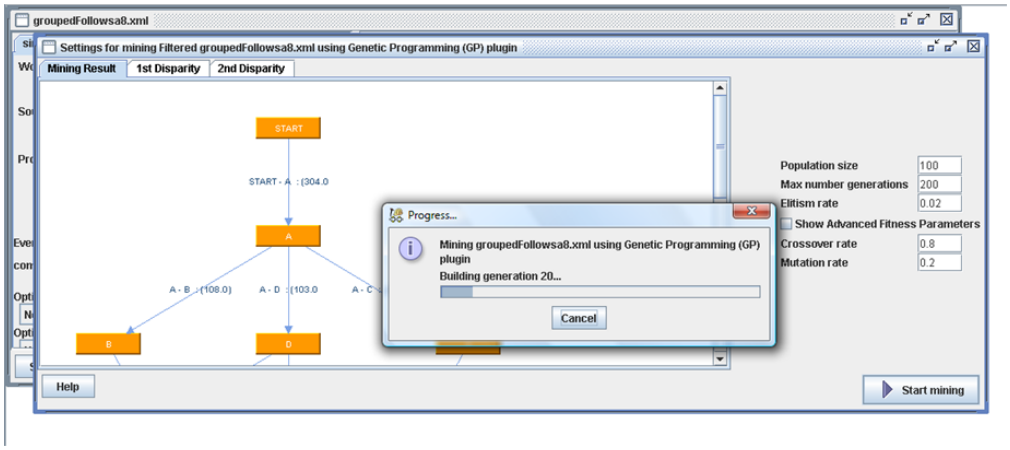

Figure 1: GUI of the Genetic Programming BPM tool.

2 Business Process Mining: Techniques and Tools

Professional Business Process Mining tackles the problem of mining process trace gathered from very large data bases. Traces, also called event logs, are data structures that describe a sequence of steps through a process. Traces can contain data about sequential or concurrent (parallel) processes. Traces can be linear or recurrent, i.e. they may contain loops. In practice, trace data can also be imperfect. Some trace data may have been entered only partially or incorrectly or some data is lost due to various causes. Data loss, for example, can appear when data is transferred between large data bases or due to formatting errors when data is standardised throughout a company.

Business processes can be described by general graphs, where each vertex or node of the graph represents a process step while each edge connects one or several nodes indicating a process transition.

The challenge in real-world Business Process Mining is the mapping of concurrent processes. Also the mining of processes with missing data or mining of highly unstructured processes where only sections of the whole process are described by the traces is difficult. Mining sequential processes is a comparatively trivial problem while building complete graphs structures from concurrent processes where only sections are covered by traces is a difficult graph matching problem. Only the combination of all traces reveils the complete branching and joining structure of the whole process graph.

Finding a concise (small) and sound workflow model from traces is a computationally complex model discovery problem. Since the mid-nineties several groups have been working on techniques for process mining [1]. The α-algorithm [2] is an example of a technique which can deal with concurrent process steps. However, this algorithm has problems with complex graph constructs and noise [3]. In fact, the general problem of finding a minimum finite-state acceptor compatible with given trace data is NP-hard [4]. Due to this complexity less precise but more robust algorithms are needed to solve these hard problems. Typically, a good choice for solving complex combinatory problems is to use Evolutionary Algorithms (EA).

While finding an optimal solution is already very difficult, in real-world applications it is also important to provide the user with a simple and easy to read and interpret visualisation of the solution. Generating a simplified but yet correct process map is a key issue for the applicability of BPM results [5]. Also an easy to interpret analysis of process disparities, i.e. the identification of deviations from given standard processes, can be very useful. Process disparities are traces that appear repeatedly though rare and which show a structure that does not completely match the given graph structure. Checking for process disparities is also referred to as conformance tests. Outliers can be indicators for fraudulent process shortcuts or clever process optimisation solutions.

Most of the commercially available process mining software cannot deal with noise, i.e. missing data or garbage left in the logs. Only few systems provide a means for analysing process disparities. However, many processes can deal with concurrent processes and even process loops.

This is a non-comprehensive list of names of the best known commercially available software products together with the names of the manufacturers: BPM|one Process Mining (Pallas Athena, Australia), Futura Reflect (Futura Process Intelligence, The Netherlands), Interstage Automated Process Discovery (Fujitsu, Japan), Comprehend (OpenConnect, US), Process Discovery Focus (Iontas, US), ARIS Process Performance Manager (IDS Scheer AG, Germany), Enterprise Visualization Suite (BusinesScape, Norway).

3 Genetic Programming Approach

Most BPM software uses adjacency matrices to describe process graphs. In Adjacency Matrices (AM) (also called causal matrices) all graph vertices are listed along the two dimensions of a symmetric matrix. AM encode very efficiently that a node 'Y' is following a node 'X'. In the matrix the (Y, X) location is set to '1', if 'Y' follows node 'X'; otherwise it is set to '0'. However, this description is very awkward to read for engineers who are more familiar with e.g. Petri Nets or workflow diagrams.

Although adjacency matrices suit the binary representation of the genomes as used in Genetic Algorithms [6] well, manipulating graph structures directly can be a lot more advantageous for programming. Working with an explicit graph structure helps identifying particular features of the mining software by the human programmer a lot faster. For developing powerful optimisation software it is important to choose a problem representation that suits the optimisation algorithm best. One should also always consider rapid fitness function evaluations as well as easy to use and human interpretable results especially when evolutionary algorithms are concerned.

An example of a sequential and a concurrent process using a Petri Net format is shown in Figure 2. The sequential process on the left follows the sequential trace: Start $> A > C > H > I > K > L >$ End. The term $'X > Y'$ indicates that $'X'$ is the predecessor of $'Y'$. On the right hand side, the concurrent process follows the trace: Start $> A > ((C > H > I > K)||(B > D > E > J)) > L >$ End, where the term $'X||Y'$ indicates that $'X'$ and $'Y'$ are executed in parallel. X and Y can either be one single step or a collection of a group of steps.

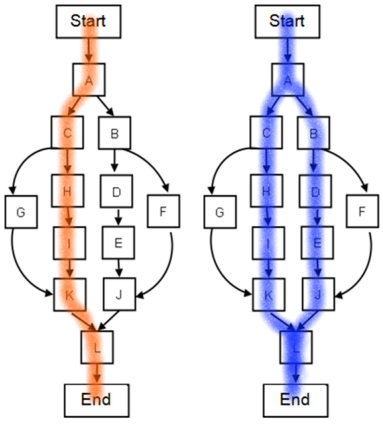

Figure 2: A sequential trace (left) and a concurrent trace (right).

The corresponding adjacency matrix for the graph from Figure 2 would consist of a sparsely filled 12×12 matrix, which is much harder to compare visually with a set of traces than the actual graph structure.

Genetic programming (GP) [7] keeps the features and advantages of genetic algorithms while being able to manipulate graph structures directly. Similar to evolutionary algorithms, GP keeps a population of solutions. Each graph structure (here also called individual) is a potential solution to the mining problem. GP compares all individuals against the traces and allocates a quality value depending on how close the graph matches the given traces.

Typically, GP has been used to manipulate tree structures e.g. for symbolic regression. However, GP is more powerful. It can vary and optimise any graph structure using genetic operators such as mutation and recombination.

In Genetic Programming of BPM graphs as introduced here, GP uses *mutation* by replacing the semantics of a logical

node in the graph. Logical nodes indicate whether two sub-processes should be executed in parallel or as two alternative processes. For example in Figure 2 the gate labelled 'A' will decide whether a process token will proceed either via 'B' or via 'C' (as shown on the left part of Figure 2) or whether 'B' and 'C' are the start points of two parallel process flows. Any gate contains either an 'AND' logic in case of parallel processed or an 'XOR' logic in case of alternative processes. GP mutation swaps the logic in a gate. Of course, GP can mutate several gates in parallel. Mutation is a random process and is applied in GP only with an (externally defined) probability. An example of a mutation of a graph structure in shown in Figure 3. Figure 3 uses a workflow net representation which shows the logic (circles) of the net [1].

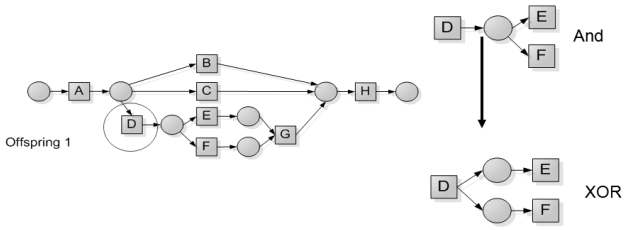

Figure 3: Mutation of gate 'D' from 'AND' to 'XOR'.

Genetic Programming can also vary the whole structure of a graph by *recombination*. Figure 4 shows a combination of two graphs thus generating a new potential BPM solution. The GP selects sub-graphs between two vertices from two individuals and swaps the sub-graphs. This is motivated by the biological crossover of genes. In this GP implementation the recommended crossover rate follows an externally user defined probability. In Figure 4 one can see that the new individual is essentially parent 2 taking over the pruned section from parent 1 (simplified connection between 'B' and 'H').

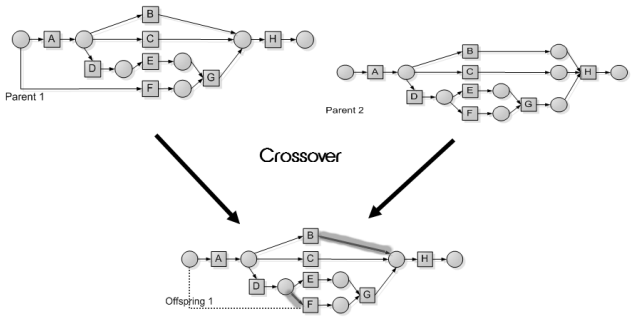

Figure 4: Recombination of two graph structures yielding a new BPM solution.

The fitness function calculates the similarity between the trace and each proposed graph structure. The first part of the fitness function measures precision, i.e. the number of additional edges present in the individual that are not recorded in the event log. The second part measures the completeness of a graph. It calculates how many of the event log traces could be parsed correctly by an individual. More details on the fitness function can be found in [8].

The GP software introduced here uses the powerful and open source software environment ProM [9]. ProM supports the implementation of process mining techniques in a standard environment. Several commercial software use ProM as an environment and come as plug-ins. Here, ProM software has been used mainly for reading traces in MXML (Mining XML) file format and for visualising the mined process graphs.

4 Example Applications

4.1 Benchmark results

To demonstrate the quality of the GP approach systematic experiments have been performed. Typical parameter settings for the Genetic Programming BPM approach that apply to most of the benchmark tests are:

1. Population size - 100
2. Mutation rate - 0.2
3. Crossover rate - 0.2
4. Max Num Generations - 200

Of course, the best parameter settings will vary a bit from application to application. In case one needs almost perfect parameters, there exist approaches using advanced statistics to optimise the parameters of the optimiser (see for example [10]).
For testing the GP approach there are standard benchmark test available in literature. For the example in Figure 5 benchmark test data from [11] is used. This dataset comes with the ProM environment. Figure 5 shows a result of a challenging business process. The process contains strong parallel (G,H,E,F) as well as asymmetric features (AB,C) which are difficult to mine.

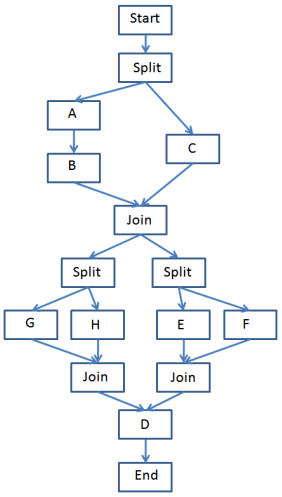

Figure 5: GP best mined result on a dataset provided by [11] (structure view).

The GP process mining technique outlined here is shown to be capable of mining both structures and semantics [8]. However, the semantics, i.e. mining the logics (AND and XOR) provide an additional level of mining difficulty. GP can also cope with more complex sequences and parallel structures (including complex parallel structures that are difficult to mine correctly by other approaches such as the GA technique ([12])).

4.2 Example of a real-world application

The proposed GP approach has also been applied to real-world applications, for example in the telecommunication industry. Here the algorithm helped in identifying the process flows in maintenance situations. The process steps of several engineers was recorded and analysed by the GP BPM. The process map includes about 40 steps and showed several decision making features (splits), where the engineer had to decide to follow either one or another process path. The BPM GP process miner was able to identify the process graph from the traced logs and could also identify process paths which lead to erroneous decisions of the engineers.

6 Transition from Academia to Industry

When software has reached a certain degree of maturity there are several ways of exploiting it commercially. A necessary first step in this direction is the evaluation of the market and estimating the market demand. The studies revealed, for example, that the overall size of the worldwide Business Process Management market for Business Process Mining was $\$1.2$ billion in 2005 and is expected to be $\$2.7$ billion by 2009 [13].

Potential routes to commercialisation of an academic software are:
1. Licensing
2. Consultancy
3. Collaboration with commercial partners
4. Setting up a spin-off company

All these options will depend on the copyright regulations of the University and the degree of commitment of all participants in the commercialisation process of the software. Setting up a full spin-off company obviously demands the highest commitment of the team which can be very significant. For running a company successfully one needs to consider all business aspects from advertisement, marketing etc. to maintenance. Going this route should be well thought through. However, this route is the most profitable but also financially the most risky way of commercialisation of research results. The route via engaging with commercial partners has the advantage that existing infrastructure coming from the partner becomes available. However, this comparatively low risk route could have the disadvantage that the actual know-how leaves the University for a comparatively low financial return. The know-how will leave especially if the software has to be re-implemented within the software environment of the client. Re-implementation will certainly help remedying any copyright issues that might come with academic software. This issue applies especially when licensing is considered. Consultancy might often be the least profitable though lowest risk approach.
Software can and should always be copyright protected. This can be done by adding standardised copyright protections clauses into the code. There are various licensing modes. Before any commercialisation is considered, the corresponding regulations within the respective country should be thoroughly studied. Consultants may help. Sometimes also Universities provide excellent consultancy.
Easy to use graphical user interfaces make software a lot better to use and sell. Providing a user-friendly GUI (see Figure 1) is not an academic exercise. However, it is a necessary step when software is going to be used by a wider community. Also providing a well maintained and structured and highly modular code is essential for commercial software. Changes to the code should be as effortless as possible. Any changes should be well documented. Professional software also comes with an easy to read and useful handbook or user manual.
Commercialisation of Evolutionary Algorithms can be particularly difficult. EA will generally yield only approximations of the perfect solution. In practice, approximations are perfectly all right as long as the solution is a significant improvement to the current *status quo*. In computer science EA are called optimisers. However, in practice they should be considered as 'improvers'. Especially EA can be extremely helpful in providing a fresh view for advanced industrial problem solving.

7 Conclusion

Genetic Programming is a powerful tool for generating process workflow maps from event logs. Genetic Programming has the advantage that it can work directly on graph structures. This makes the design of advanced Business Process Mining software a lot easier.

Often academic software gets stuck in a prototype stage. Commercialisation of academic products needs entrepreneurial thinking and skills. However, there are several options academics can choose that might end in a successful placement of good software in a market that is eagerly waiting for new and powerful products.

7 ACKNOWLEDGMENTS

This paper has been supported by the EPSRC via the project EP/G005451/1.

REFERENCES

[1] W. van der Aalst. Process Discovery: Capturing the Invisible. *IEEE Computational Intelligence Magazine*, February 2010:26–41, 2010.

[2] W. van der Aalst, A. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[3] B. van Dongen and W. van der Aalst. Multi-phase mining: Aggregating instances graphs into EPCs and Petri Nets. In *Proc. 2nd Int. Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management*, pages 35–58. USA: Florida Int. Univ., 2005.

[4] E. Gold. Complexity of automaton identification from given data. *Inform. Control*, 37(3):302–320, 1978.

[5] W. van der Aalst and C. Günther. Finding Structure in Unstructured Processes: The Case for Process Mining. In *Seventh International Conference on Application of Concurrency to System Design (ACSD 2007)*, pages 3–12. IEEE, 2007.

[6] Th. Bäck, D. B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. Oxford University Press and Institute of Physics Publishing, New York, Bristol, 1997.

[7] J. R Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

[8] Ch. Turner, A. Tiwari, and J. Mehnen. Mining Process Flowcharts From Business Data: An Evolutionary Approach. In *6th International Conference on Digital Enterprise Technology*, pages 1069–1087. Hong Kong, 14 - 19 December 2009.

[9] W. van der Aalst, B. van Dongen, C. Gnther, A. Rozinat, H. Verbeek, and A. Weijters. ProM: The Process Mining Toolkit. In *In Proceedings of the BPM 2009 Demonstration Track*, volume 489. CEUR-WS.org, 2009.

[10] Ch. Turner, A. Tiwari, and J. Mehnen. Mining Process Flowcharts From Business Data: An Evolutionary Approach. In *6th International Conference on Digital Enterprise Technology*, pages 1069–1087. Hong Kong, 14 - 19 December 2009.

[11] J. Herbst. *Ein induktiver Ansatz zur Akquisition und Adaption von Workow-Modellen*. PhD thesis, Universitat Ulm, Ulm, Germany, 2001.

[12] A. de Medeiros. *Genetic Process Mining*. PhD thesis, Eindhoven Technical University, Eindhoven, NL, 2006.

[13] K. Vollmer and C. Moore. Demand For Business Process Management Suites Will Accelerate Through 2009. *Forrester Report*, 2006.